PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

John Aram SAFA

Art Unit: 2185

Application No: 10/666,411

Examiner:

Filed: September 19, 2003

For: SOFTWARE PROTECTION

TRANSMITTAL OF CERTIFIED COPY

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

This application claims priority of United Kingdom Patent Application No. 0221984.8 filed September 21, 2002. A certified copy of the United Kingdom patent application is transmitted herewith in order to complete the claim for priority.

Respectfully submitted,

John Smith-Hill
Reg. No. 27,730

SMITH-HILL & BEDELL, P.C.
12670 N.W. Barnes Road, Suite 104
Portland, Oregon 97229

(503) 574-3100

Docket: SWIN 2793
Postcard: 12/03-48

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on the 23rd day of December, 2003.

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed J Evans.

Dated 30 September 2003

Patents Form 1/77

Patents Act 1977
(Rule

The Patent Office

1/77

THE PATENT OFFICE
D
21 SEP 2002
NEWPORT

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference MPS/8227

22SEP02 E750120-7 D02833
700 0.00-0221984.8

2. ~~Patent application number~~
(The Patent Office will fill in this part) 0221984.8

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

BitArts Limited
3rd Floor, 15 Middle Pavement,
Nottingham, NG1 7DX.

Patents ADP number *(if you know it)* 8077232002

If the applicant is a corporate body, give the country/state of its incorporation United Kingdom

4. Title of the invention

Software Protection

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Swindell & Pearson

48 Friar Gate,
Derby DE1 1GY

Patents ADP number *(if you know it)* 00001578001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*

a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

Patents Form 1/77

Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	0
Description	14
Claim(s)	7
Abstract	0
Drawing(s)	3

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature Swindell & Pearson Date 20/09/02

Swindell & Pearson

12. Name and daytime telephone number of person to contact in the United Kingdom: Mr. M.P. Skinner - 01332 367051

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

Software Protection

The present invention relates to software protection arrangements.

Protection arrangements are necessary for software to prevent unlicensed copies of commercial software being made and distributed among users. This deprives the proprietor of the software from legitimate income from the sale of licences.

The present invention provides a software protection arrangement including:

identifying means operable to create an identifier which characterises the machine on which the protected software is to be run;

authorisation means operable to receive an identifier created by the identifying means to execute a predetermined function on a received identifier to form a derived identifier, execution of the predetermined function being conditional upon verification of a condition required for authorisation of the use of the software;

and the arrangement further comprising enabling means operable to enable execution of the protected software only when in receipt of an enabling identifier from the authorisation means, the derived identifier serving as an enabling identifier in the event that the derived identifier has been derived by the predetermined function from the identifier of the machine on which the protected software is to be run.

The enabling means may be operable to apply a function to the derived identifier to recover the identifier from which the derived identifier was derived, and to compare the recovered identifier with the identifier created by the identifying means, and to enable or disable execution of the software in accordance with the result of the comparison.

Preferably the protected software is in encrypted form requiring decryption by at least one decryption key for successful execution, the enabling

means including decryption means operable to execute a process which includes decryption of the encrypted code, and to use the derived identifier as a key for the process.

Preferably the predetermined function is a function of at least two variables, a received identifier forming one of the variables, and the other variable being a confidential decryption key stored at the authorisation means, and wherein the enabling means is operable to perform a preliminary step to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential decryption key for use as a decryption key in decrypting the encrypted code.

Preferably the identifier further includes information characterising the protected software, and the authorisation means is operable to select a confidential decryption key corresponding with the identified software.

Preferably the identifier is derived from information which identifies hardware and/or software present at the machine.

The authorisation means may be operable to effect a financial transaction or credit check before allowing execution of the predetermined function.

Preferably, the identifying means is operable to create an identifier as aforesaid on each occasion protected software is to run.

The arrangement may further comprise transmission means by which the identifying means transmits identifiers to the authorisation means. The transmission means may comprise a communication network. The authorisation means may be operable to transmit derived identifiers to the enabling means by means of the transmission means.

The enabling means and/or the identifying means are preferably

provided by software elements associated with the protected software.

In a second aspect, the invention provides an arrangement for use in protecting software, the arrangement including:

identifying means operable to create an identifier which characterises the machine on which the protected software is to be run;

enabling means operable to receive a derived identifier derived by authorisation means from the identifier created by the identifying means, and the enabling means being further operable to enable execution of the software only when in receipt of an enabling identifier, the derived identifier serving as an enabling identifier in the event that the derived identifier has been derived by the predetermined function from the identifier of the machine on which the software is to be run.

The enabling means may be operable to apply a function to the derived identifier to recover the identifier from which the derived identifier was derived, and to compare the recovered identifier with the identifier created by the identifying means, and to enable or disable execution of the software in accordance with the result of the comparison.

Preferably the protected software is in encrypted form requiring decryption by at least one decryption key for successful execution. The enabling means may include decryption means operable to execute a process which includes decryption of the encrypted code, and to use the derived identifier as a key for the process.

Preferably the derived identifier is derived by a predetermined function which is a function of at least two variables, a received identifier forming one of the variables, and other variable being a confidential decryption key stored at the authorisation means, and wherein the enabling means is operable to perform a preliminary step to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential decryption key for use as a decryption key in decrypting the

encrypted code.

Preferably the identifier further includes information characterising the protected software, whereby the authorisation means may operate to select a confidential decryption key corresponding with the identified software.

Preferably the identifier is derived from information which identifies hardware and/or software present at the machine.

Preferably, the identifying means is operable to create an identifier as aforesaid on each occasion protected software is to run.

The enabling means and/or the identifying means are preferably provided by software elements associated with the protected software.

In a third aspect, the invention provides an arrangement for use in protection of software, the arrangement including:

authorisation means operable to receive an identifier characterising a machine on which protected software is to be run, and the authorisation means being operable to execute a predetermined function on a received identifier to form a derived identifier, execution of the predetermined function being conditional upon verification of a condition required for authorisation of the use of the software; and to provide the derived identifier to allow enabling means to enable execution of the software only when in receipt of an enabling identifier which is a derived identifier derived from the identifier of the machine on which the software is to be run.

The predetermined function may be a function of at least two variables, a received identifier forming one of the variables, and another variable being a confidential decryption key stored at the authorisation means, wherein a preliminary step is required upon receipt of a derived identifier by enabling means, to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential

decryption key for use as a decryption key in decrypting an encrypted form of the protected software.

The identifier may include information characterising the protected software, the server being operable to select a confidential decryption key corresponding with the identified software.

The authorisation means is preferably operable to effect a financial transaction or credit check before allowing execution of the predetermined function.

The invention also provides computer software which, when installed on one or more computer systems, is operable to provide a software protection arrangement as set out above.

The invention also provides a carrier medium for software as defined in the previous paragraph. The medium may be a memory device or a transmission medium on which the software is carried by a propagating signal. The invention also provides a signal propagating as aforesaid. The invention also provides a signal propagating on a transmission medium and carrying an identifier or derived identifier of a software protection arrangement as defined above.

The invention also provides a method of protecting software including the steps of:

creating an identifier which characterises the machine on which the protected software is to be run;

receiving an identifier and executing a predetermined function on a received identifier to form a derived identifier, execution of the predetermined function being conditional upon verification of a condition required for authorisation of the use of the software;

and enabling execution of the protected software only in response to an enabling identifier, the derived identifier serving as an enabling identifier in

the event that the derived identifier has been derived by the predetermined function from the identifier of the machine on which the protected software is to be run.

A function may be applied to the derived identifier to recover the identifier from which the derived identifier was derived, and to compare the recovered identifier with the identifier created by the identifying means, and to enable or disable execution of the software in accordance with the result of the comparison.

Preferably the protected software is in encrypted form requiring decryption by at least one decryption key for successful execution, the enabling step including a decryption step which includes decryption of the encrypted code, the derived identifier being used as a key for the decryption step.

Preferably the predetermined function is a function of at least two variables, a received identifier forming one of the variables, and the other variable being a confidential decryption key, the enabling step including a preliminary step to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential decryption key for use as a decryption key in decrypting the encrypted code.

Preferably the identifier is created to include information characterising the protected software, and the confidential decryption key is selected according to the software identified.

Preferably the identifier is derived from information which identifies hardware and/or software present at the machine.

Preferably a financial transaction or credit check is effected before allowing execution of the predetermined function.

Embodiments of the present invention will now be described in more detail, by way of example only, and with reference to the accompanying drawings, in which:

Fig. 1 illustrates a general purpose computer by means of which the present invention may be implemented;

Fig. 2 illustrates part of a computer of the type illustrated in Fig. 1 and on which protected software is to be run;

Fig. 3 illustrates part of a server based on a computer of the type illustrated in Fig. 1;

Figs. 4a and 4b illustrate simplified sequences of steps for enabling execution of the protected software; and

Figs. 5a and 5b respectively illustrate the locations at which the various steps of the sequences of Figs. 4a and 4b, respectively, take place.

Hardware Arrangement

Fig. 1 illustrates a general purpose computer 10 by means of which the present invention may be implemented. The computer 10 may be, for example, an IBM compatible personal computer (PC) running under appropriate software control. Alternatively, the computer 10 may be a computer of alternative design.

In Fig. 1, the computer 10 includes a central processor 12 with associated main (RAM) memory 14 and auxiliary memory 16 in the form of a hard disc drive. A display screen 18 and keyboard 20 are provided for use by a user. Other conventional input and output arrangements may be provided at 22, preferably including a device for reading a portable memory medium such as a floppy disc 24, by means of which software and/or data may be loaded into or

out of the computer 10. The components described above are in communication with each other by virtue of a data bus 25 to which they are connected. The data bus 25 is also connected to an external communication link 26, such as a connection to the internet or other private or public communication network. The link 26 may be effected by means of a modem.

A skilled reader will have no difficulty in obtaining appropriate hardware and software to form a general purpose computer of the type described above and suitable for implementing the present invention, once the description set out below has been fully understood.

Machine on which the Software is to be Run

Fig. 2 shows part of a machine 10A on which protected software is to be run. The machine 10A may be a general purpose computer of the type illustrated in Fig. 1. The computer 10A is shown only in part in Fig. 2, in the interests of clarity.

Within the machine 10A, the processor 12A, RAM 14A, hard drive 16A and modem 26A each has associated with it a unique identifier, which enables the component to be distinguished from other otherwise identical hardware components. The identifier 28 will be permanently built into the component during manufacture.

The hard disc 16A stores a copy of the protected software 30, ready for loading into the RAM 14A for execution, under control of the security arrangements. The RAM 14A has an area 32A containing the operating system and an area 34A available for loading application software. The area 34A is shown as containing three software elements, namely an identifying module 36, an enabling module 38 and an executable form 40 of the software 30. The executable form 40 is shown in broken lines to indicate that its availability is dependent on the security arrangements being described.

Server

Fig. 3 illustrates an authorisation arrangement embodied in this example as a server remote from the machine 10A. Components of the server which correspond with the components shown in Fig. 2, bearing the corresponding numeral and the suffix B.

Within the server, the RAM 14B is shown as containing two software modules in addition to the operating system 32B, namely a module 42 operable to execute a predetermined function, and a finance or credit checking module 44.

The hard disc 16B may include the data of one or more databases for access by the modules 42, 44 as required, as will become apparent.

In this example, the server operates to execute automatically the authorisation functions. In alternative embodiments, the authorisation arrangement can be embodied in other ways. For example, software modules could be provided within the machine 10A to perform the authorisation functions to be described. Alternatively, the authorisation functions could be provided remotely, but not automatically, or semi-automatically. For example, communication between the machine 10A and the authorisation arrangement could involve steps taken by a human operator, such as a telephone message, or the authorisation arrangement could involve a human operator operating a machine or otherwise providing the authorisation functions.

Functions of the Modules

The functions of the various software modules can be illustrated as a sequence of steps as shown in Figs. 4a and 4b. Figs. 5a and 5b illustrate more graphically the location at which these steps are implemented.

In both embodiments, the identifying module 36 executes, preferably on

each occasion software is to be run, to create an identifier which includes information characterising the machine on which the software is to be run. This identifier is created by interrogating various components of the machine 10A to determine their component identifiers 28, and combining one or more of these identifiers to create an identifier which includes information characterising the machine 10A. The identifier may be created by combining one or more identifiers 28 by an algorithm of any desired complexity. This algorithm is illustrated at 46 as f (hardware) to indicate a function applied to hardware identifiers 28. In Fig. 4a, function f (hardware) returns the value 1234. It is to be understood that this represents only an example. The value returned will depend on the identifiers 28 forming the arguments of the function, and thus will depend on the machine on which the module 36 is being executed. The value returned could be alpha-numeric or a binary string or recorded in other machine readable form and the length of the identifier could vary from that shown, according to the nature of the algorithm f.

In this example, the identifier 1234 is sent by means of the modem 26 to the server 10B. Alternatively, the identifier could be sent internally of the machine 10A to the authorising means, or externally by human intervention. The authorisation means, in this case the server 10B, receives the identifier from the machine 10A and operates on it by means of the predetermined function module 42. In this example, the module 42 applies a function illustrated as g, at 48, to return a value derived from the received identifier (1234 in this example) and here called the derived identifier. In this example, and purely for purposes of example, the derived identifier is shown as WXYZ. Thus, g (1234) = WXYZ.

It will be clearly apparent that the value of the derived identifier depends on the value of the received identifier, and on the nature of the function g.

Prior to execution of function g, verification is required in order to ensure that it is appropriate to authorise the protected software to be used. Verification involves the verification of a condition required for authorisation.

For example, the condition may be financial, in which case, the finance or credit check module 44 is called. This serves to identify the machine 10A from the received identifier, perhaps in conjunction with a database in the hard disc 16B. A financial transaction may then be executed, such as a debit to a credit card account, or a credit check may be made before passing control back to the function module 42 for execution of the function g. Alternatively, the module 44 may verify that the protected software is authorised for use on the identified machine.

The use of a finance or credit check is optional and may not always be required or desirable. However, the use of a module 44 will always be required in order to effect verification of a condition, and only to authorise execution of the function g in the event that the result of verification is positive. Consequently, the ? symbol is associated with the connections between the functions f and g in Figs. 4A and 5A.

The derived identifier WXYZ is transmitted back to the machine 10A, preferably over the same communication network, i.e. by means of the modems 26.

The derived identifier serves as input to the enabling module 38 which, in this example, executes a further function h on the derived identifier, at 50. The function h is devised to recover the identifier 28 from the derived identifier. Thus, h (WXYZ) = 1234. Function h is the inverse of function g.

The enabling module 38 concludes by making a comparison at 52 between the result of function h applied to the derived identifier, and the identifier created by the module 36 and sent to the machine 10B. These will be identical in the event that the identifier and derived identifier have been sent from and to the same machine, and that the sending of a derived identifier has been authorised by the module 44.

If use of the software is not authorised, no derived identifier will be

received. If a received identifier is used with a different machine, the comparison will fail. The enabling module 38 is programmed to prevent execution of the software 30 in the absence of a derived identifier, or the failure of the comparison. The software 30 is thus protected from execution except on a single authorised machine.

Second Embodiment

In this example, the first step at 46 is again to create an identifier by interrogating the identifier 28 of the constituent components of the machine 10A. Again, this is illustrated as returning the value 1234. This step is executed by the identifying module 36. The identifier is sent to the authorisation means, again in the form of a server 10B, by means of the modem 26.

In this example, the software 30 is held in encrypted form in the hard disc 16A, and the enabling module 38 is required to decrypt by using a decryption key. The decryption key is created as follows.

At the server 10B, the identifier created by the module 36 is received and used at 54 as a variable for a predetermined function j. Function j is authorised to execute only upon verification of a required condition, such as a satisfactory financial transaction or check, as described above. Consequently, the ? symbol is again used in Figs. 4B and 5B.

Function j is a function having at least two variables. In this example, the second variable is shown as ABCD, which is a confidential decryption key stored at the server, in the hard disc 16B.

In a simple form of this example, the same confidential decryption key will be used on each occasion. In a more complex arrangement, a range of confidential decryption keys may be available to the machine 10B. For example, the received identifier may further include information characterising the

protected software, the module 42 selecting a confidential decryption key corresponding with the software identified by the identifier. Thus, all encrypted copies of a particular application could be associated with the same confidential decryption key, there being a different confidential decryption key associated with all encrypted copies of a different application.

Having selected the appropriate confidential decryption key ABCD, the module 42 executes function j, returning the value MNOP, i.e. j (1234, ABCD) = MNOP.

MNOP forms the derived identifier, being derived, in part, from the identifier 1234. The derived identifier MNOP is sent back to the computer 10A, preferably by means of the modems 26.

The derived identifier MNOP is received by the enabling module 38 which, in this example, first executes a preliminary step at 56 by applying a second predetermined function k to the received identifier. Function k is a function of at least two variables, one being the derived identifier MNOP, and the other being the identifier created by the module 36. Function k is chosen such that by applying this to the variables MNOP and 1234, the confidential decryption key supplied within the computer 10B is returned. Thus, k (MNOP, 1234) = ABCD.

The value returned from function k is then used as a decryption key at 58 by the enabling module 38, to decrypt the software copy at 30, for execution at 40.

If use of the protected software is not authorised for the machine sending the identifier, no derived identifier is returned and the software cannot be decrypted. If function k is executed on a machine which is not the machine from which the derived identifier MNOP was ultimately derived, the identifier used will be incorrect and the result of function k will not be the correct value ABCD. Consequently, the decryption of the software 30 will fail. Similarly, if

the derived identifier has been derived from the incorrect confidential decryption key, decryption will again fail.

It is also to be noted that the decryption code ABCD has been made available within the machine 10A for decryption, but without being sent across the communication network. In effect, an encrypted encryption key is sent, so that these two layers of encryption improve the protection provided to the software 30.

Alternative Arrangements

It will be readily apparent to the skilled reader that many alternatives can be devised for the arrangements described above. The various functions which have been described could be of arbitrarily great complexity, subject to the availability of appropriate processing power. The various functions described can be implemented in various combinations of hardware and software. Many different examples of appropriate technologies could be chosen for the hardware items described.

The various software modules described above can be carried on a carrier medium prior to installation, such as on a memory device or as a signal propagating on a transmission medium.

Whilst endeavouring in the foregoing specification to draw attention to those features of the invention believed to be of particular importance it should be understood that the Applicant claims protection in respect of any patentable feature or combination of features hereinbefore referred to and/or shown in the drawings whether or not particular emphasis has been placed thereon.

CLAIMS

1. A software protection arrangement including:

identifying means operable to create an identifier which characterises the machine on which the protected software is to be run;

authorisation means operable to receive an identifier created by the identifying means to execute a predetermined function on a received identifier to form a derived identifier, execution of the predetermined function being conditional upon verification of a condition required for authorisation of the use of the software;

and the arrangement further comprising enabling means operable to enable execution of the protected software only when in receipt of an enabling identifier from the authorisation means, the derived identifier serving as an enabling identifier in the event that the derived identifier has been derived by the predetermined function from the identifier of the machine on which the protected software is to be run.

2. An arrangement according to claim 1, wherein the enabling means is operable to apply a function to the derived identifier to recover the identifier from which the derived identifier was derived, and to compare the recovered identifier with the identifier created by the identifying means, and to enable or disable execution of the software in accordance with the result of the comparison.

3. An arrangement according to claim 1, wherein the protected software is in encrypted form requiring decryption by at least one decryption key for successful execution, the enabling means including decryption means operable to execute a process which includes decryption of the encrypted code, and to use the derived identifier as a key for the process.

4. An arrangement according to any preceding claim, wherein the predetermined function is a function of at least two variables, a received identifier forming one of the variables, and the other variable being a

confidential decryption key stored at the authorisation means, and wherein the enabling means is operable to perform a preliminary step to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential decryption key for use as a decryption key in decrypting the encrypted code.

5. An arrangement according to any preceding claim, wherein the identifier further includes information characterising the protected software, and the authorisation means is operable to select a confidential decryption key corresponding with the identified software.

6. An arrangement according to any preceding claim, wherein the identifier is derived from information which identifies hardware and/or software present at the machine.

7. An arrangement according to any preceding claim, wherein the authorisation means is operable to effect a financial transaction or credit check before allowing execution of the predetermined function.

8. An arrangement according to any preceding claim, wherein the identifying means is operable to create an identifier as aforesaid on each occasion protected software is to run.

9. An arrangement according to any preceding claim, further comprising transmission means by which the identifying means transmits identifiers to the authorisation means.

10. An arrangement according to claim 9, wherein the transmission means comprise a communication network.

11. An arrangement according to claim 9 or 10, wherein the authorisation means is operable to transmit derived identifiers to the enabling means by means of the transmission means.

12. An arrangement according to any preceding claim, wherein the enabling means and/or the identifying means are provided by software elements associated with the protected software.

13. An arrangement for use in protecting software, the arrangement including:

identifying means operable to create an identifier which characterises the machine on which the protected software is to be run;

enabling means operable to receive a derived identifier derived by authorisation means from the identifier created by the identifying means, and the enabling means being further operable to enable execution of the software only when in receipt of an enabling identifier, the derived identifier serving as an enabling identifier in the event that the derived identifier has been derived by the predetermined function from the identifier of the machine on which the software is to be run.

14. An arrangement according to claim 13, wherein the enabling means are operable to apply a function to the derived identifier to recover the identifier from which the derived identifier was derived, and to compare the recovered identifier with the identifier created by the identifying means, and to enable or disable execution of the software in accordance with the result of the comparison.

15. An arrangement according to claim 13, wherein the protected software is in encrypted form requiring decryption by at least one decryption key for successful execution.

16. An arrangement according to claim 15, wherein the enabling means include decryption means operable to execute a process which includes decryption of the encrypted code, and to use the derived identifier as a key for the process.

17. An arrangement according to any of claims 13 to 16, wherein the derived

identifier is derived by a predetermined function which is a function of at least two variables, a received identifier forming one of the variables, and other variable being a confidential decryption key stored at the authorisation means, and wherein the enabling means is operable to perform a preliminary step to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential decryption key for use as a decryption key in decrypting the encrypted code.

18. An arrangement according to any of claims 13 to 17, wherein the identifier further includes information characterising the protected software, whereby the authorisation means may operate to select a confidential decryption key corresponding with the identified software.

19. An arrangement according to any of claims 13 to 18, wherein the identifier is derived from information which identifies hardware and/or software present at the machine.

20. An arrangement according to any of claims 13 to 19, wherein the identifying means is operable to create an identifier as aforesaid on each occasion protected software is to run.

21. An arrangement according to any of claims 13 to 20, wherein the enabling means and/or the identifying means are preferably provided by software elements associated with the protected software.

22. An arrangement for use in protection of software, the arrangement including:

authorisation means operable to receive an identifier characterising a machine on which protected software is to be run, and the authorisation means being operable to execute a predetermined function on a received identifier to form a derived identifier, execution of the predetermined function being conditional upon verification of a condition required for authorisation of the use of the software; and to provide the derived identifier to allow enabling

means to enable execution of the software only when in receipt of an enabling identifier which is a derived identifier derived from the identifier of the machine on which the software is to be run.

23. An arrangement according to claim 22, wherein the predetermined function is a function of at least two variables, a received identifier forming one of the variables, and another variable being a confidential decryption key stored at the authorisation means, wherein a preliminary step is required upon receipt of a derived identifier by enabling means, to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential decryption key for use as a decryption key in decrypting an encrypted form of the protected software.

24. An arrangement according to claim 22 or 23, wherein the identifier includes information characterising the protected software, the server being operable to select a confidential decryption key corresponding with the identified software.

25. An arrangement according to any of claims 22 to 24, wherein the authorisation means is operable to effect a financial transaction or credit check before allowing execution of the predetermined function.

26. A software protection arrangement substantially as described above, with reference to the accompanying drawings.

27. Computer software which, when installed on one or more computer systems, is operable to provide a software protection arrangement of any of claims 1 to 26.

28. A carrier medium for software as defined in claim 27.

29. A medium according to claim 28, the medium being a memory device or a transmission medium on which the software is carried by a propagating

signal.

30. A signal propagating on a transmission medium and carrying software as defined in claim 27.

31. A signal propagating on a transmission medium and carrying an identifier or derived identifier of a software protection arrangement as defined in any of claims 1 to 26.

32. A method of protecting software including the steps of:

creating an identifier which characterises the machine on which the protected software is to be run;

receiving an identifier and executing a predetermined function on a received identifier to form a derived identifier, execution of the predetermined function being conditional upon verification of a condition required for authorisation of the use of the software;

and enabling execution of the protected software only in response to an enabling identifier, the derived identifier serving as an enabling identifier in the event that the derived identifier has been derived by the predetermined function from the identifier of the machine on which the protected software is to be run.

33. A method according to claim 32, wherein a function is applied to the derived identifier to recover the identifier from which the derived identifier was derived, and to compare the recovered identifier with the identifier created by the identifying means, and to enable or disable execution of the software in accordance with the result of the comparison.

34. A method according to claim 32, wherein the protected software is in encrypted form requiring decryption by at least one decryption key for successful execution, the enabling step including a decryption step which includes decryption of the encrypted code, the derived identifier being used as a key for the decryption step.

35. A method according to any of claims 32 to 34, wherein the predetermined function is a function of at least two variables, a received identifier forming one of the variables, and the other variable being a confidential decryption key, the enabling step including a preliminary step to execute a second predetermined function of at least two variables, including the identifier and the derived identifier, to recover the confidential decryption key for use as a decryption key in decrypting the encrypted code.

36. A method according to any of claims 32 to 35, wherein the identifier is created to include information characterising the protected software, and the confidential decryption key is selected according to the software identified.

37. A method according to any of claims 32 to 36, wherein the identifier is derived from information which identifies hardware and/or software present at the machine.

38. A method according to any of claims 32 to 37, wherein a financial transaction or credit check is effected before allowing execution of the predetermined function.

39. A method of protecting software, substantially as described above, with reference to the accompanying drawings.

40. Any novel subject matter or combination including novel subject matter disclosed herein, whether or not within the scope of or relating to the same invention as any of the preceding claims.

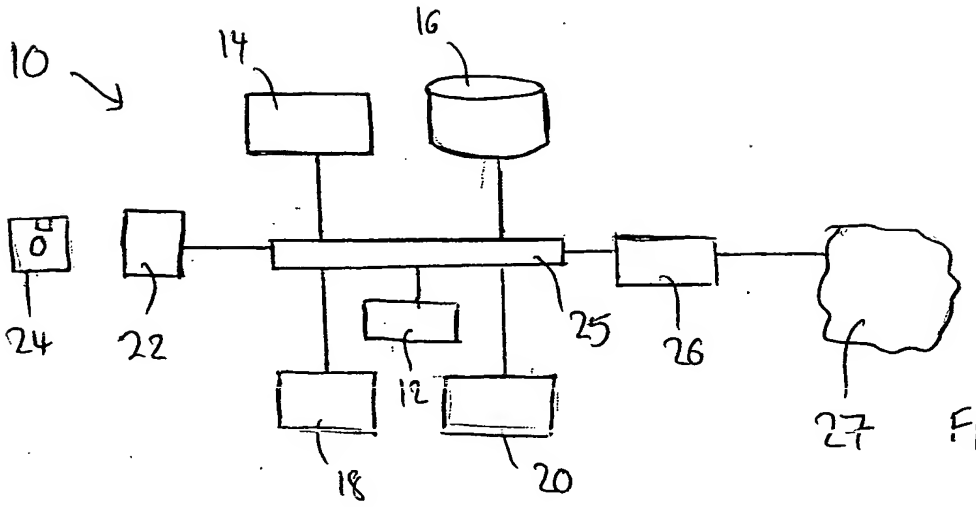
10
14
16
24
22
25
26
12
18
20
27

FIG 1

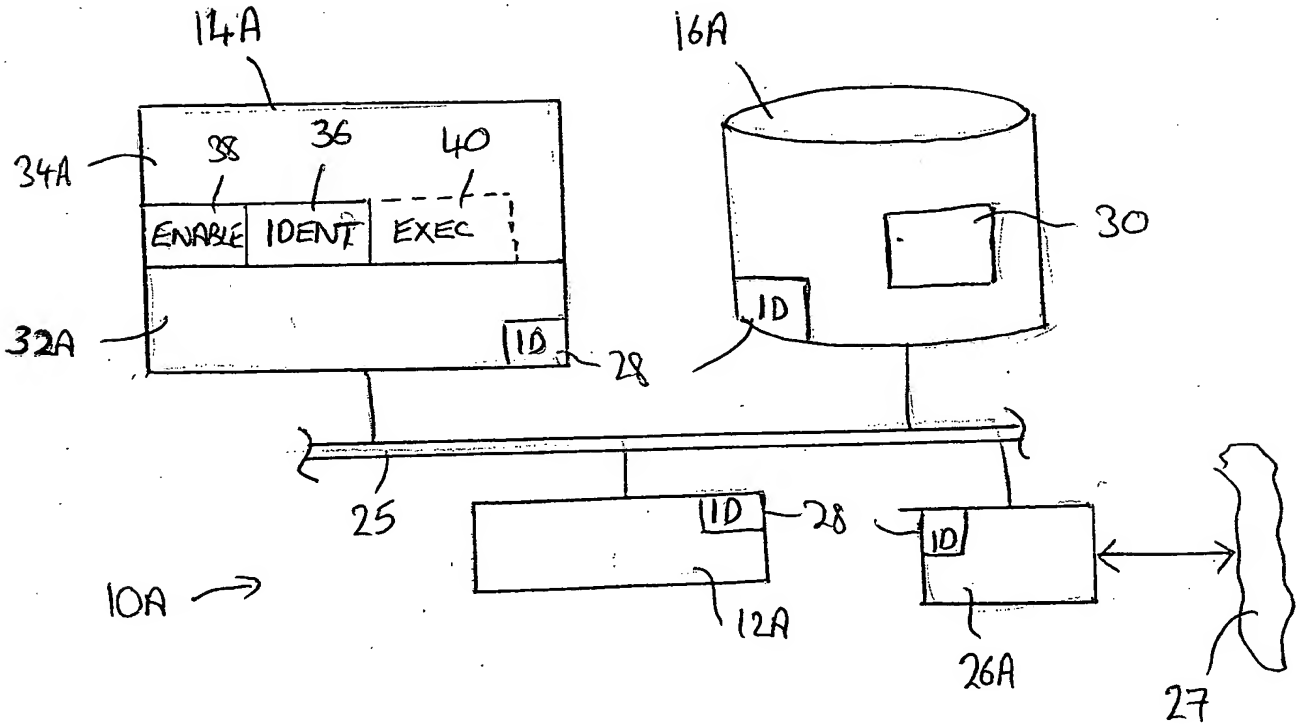
14A
16A
34A
38
36
40
ENABLE
IDENT
EXEC
32A
ID
28
30
25
10A
12A
26A
27

FIG. 2

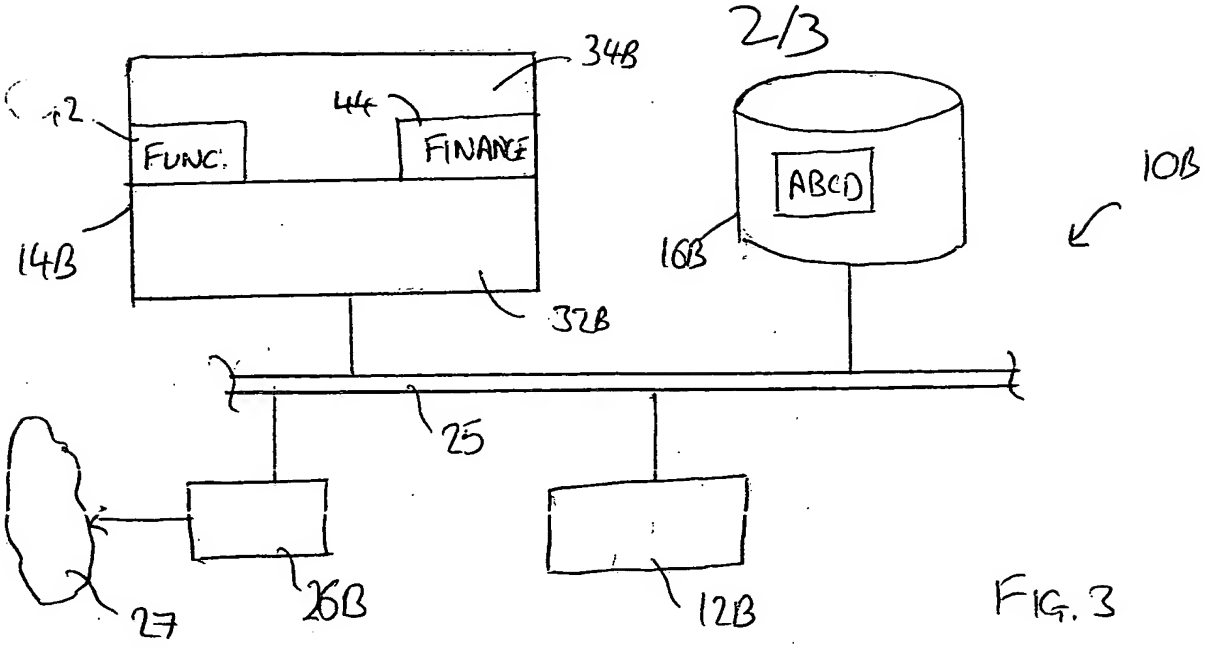

FIG. 3

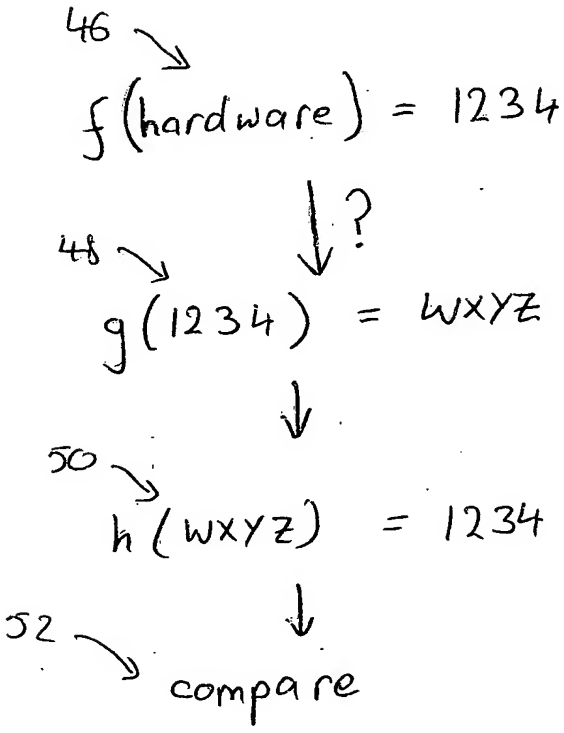

FIG 4A

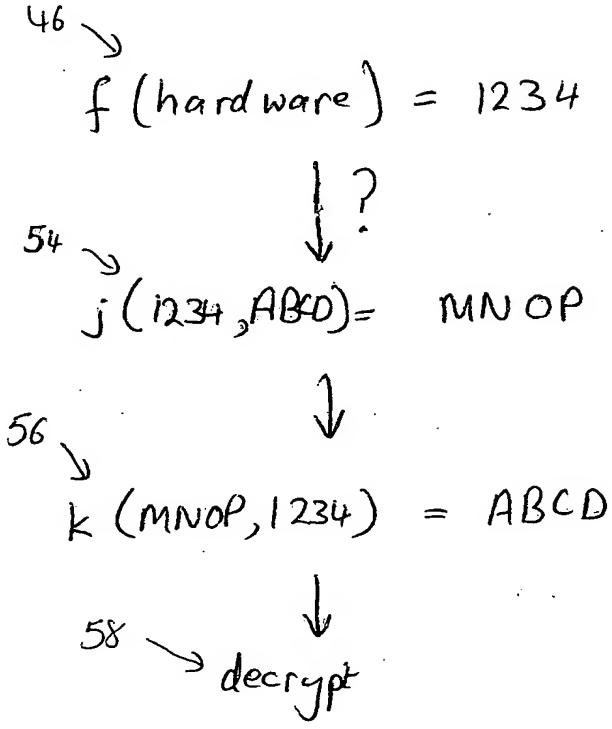

FIG 4B

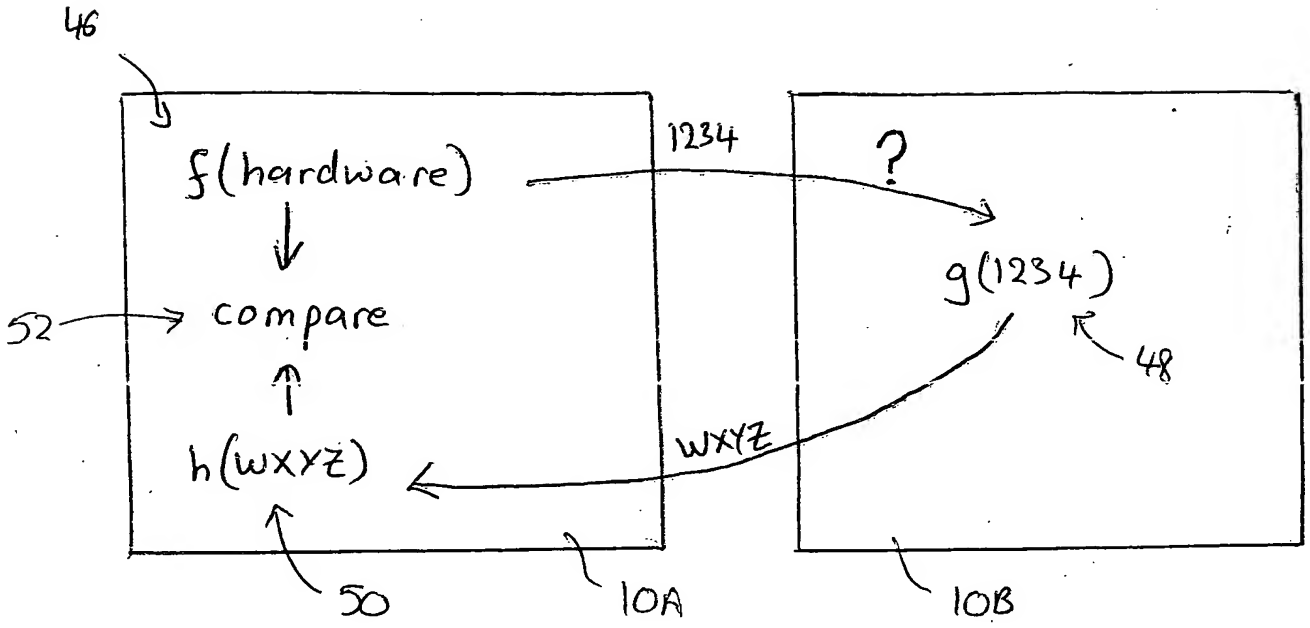
46
f(hardware)
1234
?
g(1234)
48
52
compare
WXYZ
h(WXYZ)
50
10A
10B

FIG 5A

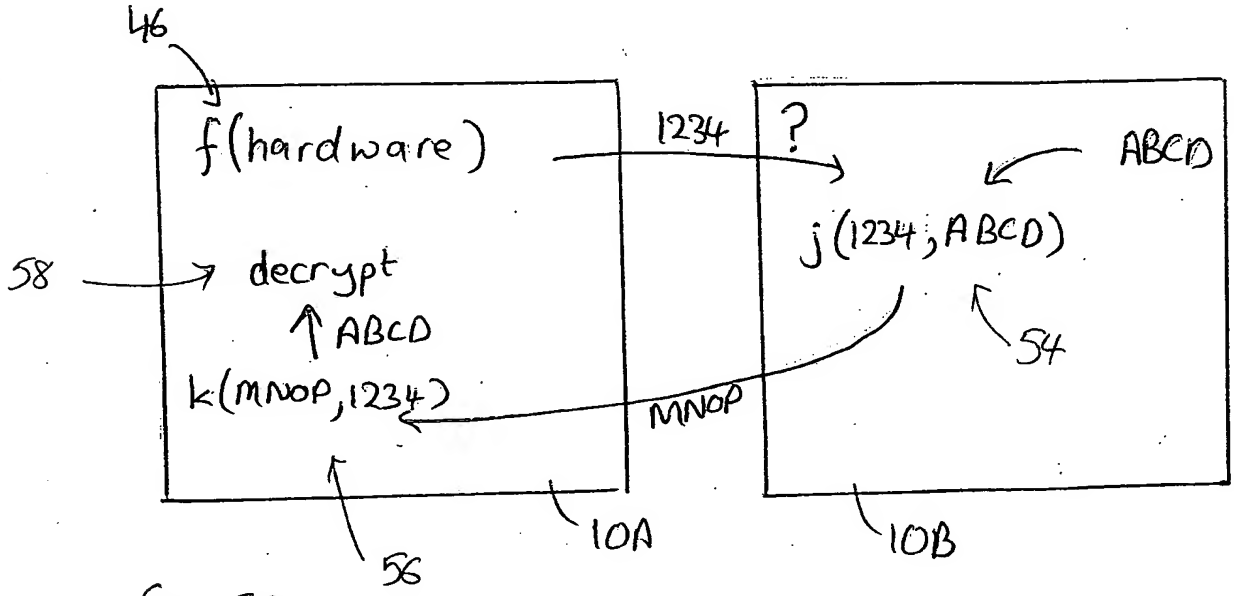
46
f(hardware)
1234
?
ABCD
j(1234;ABCD)
58
decrypt
ABCD
54
k(MNOP,1234)
MNOP
56
10A
10B

FIG 5B